

## Третье практическое домашнее задание.

Дедлайн: 22.03.2023 23:59:59

### *Требования к графу.*

Так как граф это математическая абстракция, его можно параметризовать как угодно. Более того, это множество вершин и пар вершин aka рёбер.

1. Сделать абстрактный класс граф. Параметризовать его типом вершины (по умолчанию это ее номер) и типом ребра (по умолчанию это пара вершин).
2. От графа отнаследовать две реализации: на списке смежности и на матрице смежности.
3. Граф должен уметь.
  - Отдавать число вершин и ребер.
  - Отдавать по вершине список соседей (в вершинах).
  - Отдавать по вершине итератор на список соседей. Должен работать **range-based for**.
  - (По желанию) Сделать аналог **filter iterator**, который будет способен фильтровать ребра в зависимости от условия.
4. Прописать юзинги `vertex_type`, `edge_type`.

### *Требования к взвешенному графу.*

1. Все требования к графу, кроме требования про наследника в виде матрицы смежности. Хватит на основе списка смежности.
2. Ребро должно быть взвешенным, то есть хранить в себе вес. Его тип считаем тоже шаблоном, чтобы уметь обрабатывать как целые, так и вещественные числа.

### *Требование к СНМ.*

Нужно уметь хранить множества не только из чисел, но и из произвольных типов.

### *Требования к остовным деревьям.*

1. Так как дерево является графом, оно должно быть его наследником. В случае взвешенного дерева оно должно быть наследником взвешенного графа.
2. Реализуйте оба алгоритма (Прима и Крускала) построения миностова. Это функции, которые принимают на вход граф, а возвращают дерево. Учтите, что может быть придется воспользоваться техникой [trailing return type](#).